

Università della Calabria  
Corsi di Laurea in Chimica, Fisica e Scienze dei Materiali

**Esame di Informatica**

(Variante A)

Non è consentito l'uso di alcun tipo di documentazione e strumentazione elettronica.  
Si utilizzi il retro di queste pagine per svolgere i calcoli e implementare i programmi richiesti.  
Tempo a disposizione [hh:mm]: **1:30**

Cognome: \_\_\_\_\_ Nome: \_\_\_\_\_ Matricola \_\_\_\_\_

Corso di Studi:  Chimica  Fisica  Scienze dei Materiali Data [gg/mm/aaaa]: \_\_\_\_\_

1. [2 punti] Si converta in notazione binaria il numero naturale decimale ottenuto sommando al numero 23 l'ultima cifra del proprio numero di matricola. Qualora l'ultima cifra fosse 0, sommare 5. Per esempio, se la matricola fosse 826454, convertire in binario il numero 27. Se, invece, la matricola fosse 826570, convertire il numero 28.

**Soluzione:**

2. [1 punto] È possibile rappresentare il numero intero decimale -11 in binario in complemento a due con 4 bit?
- Sì, poichè l'intervallo di variazione dell'insieme è  $[-2^4 - 1, 2^4]$   
 Sì, poichè l'intervallo di variazione dell'insieme è  $[-2^4, 2^4 - 1]$   
 No, poichè l'intervallo di variazione dell'insieme è  $[0, 2^3 - 1]$   
 **No, poichè l'intervallo di variazione dell'insieme è  $[-2^3, 2^3 - 1]$**
3. [2 punti] Il numero intero binario 11111111 in complemento a 2 con otto bit corrisponde a:
- 128  
 **-1**  
 1  
 127
4. [1 punto] In cosa consiste l'operazione di *fetch* effettuata dall'Unità di Controllo di un calcolatore di von Neumann?
- Prelevamento dalla memoria principale del prossimo dato da calcolare;  
 **Prelevamento dalla memoria principale della prossima istruzione da eseguire;**  
 Prelevamento dal registro di output dell'ALU del risultato dell'ultima operazione aritmetico-logica;  
 Prelevamento dall'Instruction Register della prossima istruzione da eseguire.
5. [1 punto] Quale delle seguenti affermazioni è vera a proposito della memoria principale di un calcolatore di von Neumann?
- Ogni cella ha un indirizzo numerico e contiene istruzioni e dati in formato binario;**  
 Ogni cella ha un indirizzo numerico e contiene istruzioni in formato esadecimale;  
 Ogni cella ha un indirizzo numerico e contiene dati in formato binario;  
 Ogni cella ha un indirizzo alfanumerico e contiene istruzioni in formato ottale;

**Esame di Informatica**  
(Variante A)

---

6. [4 punti] Si consideri la seguente successione di numeri naturali:

$$\begin{aligned}a_1 &= 1 \\ a_2 &= 1 \\ a_3 &= 2 \\ a_n &= a_{n-1} + a_{n-2} - (a_{n-3} + 1)\end{aligned}$$

Si scriva una funzione che riceva in input un intero  $n \geq 1$  e ritorni l' $n$ -esimo termine della successione. La funzione ritorni 0 se  $n < 1$ . L'algoritmo non utilizzi oggetti array per la risoluzione del problema.

**Soluzione:**

```
int termineSuccessione(const int n)
{
    if (n < 1) return 0;
    if (n == 1) return 1;
    if (n == 2) return 1;
    if (n == 3) return 2;

    int d = 1; //a_{n-3}
    int c = 1; //a_{n-2}
    int b = 2; //a_{n-1}
    int a;     //a_n
    for (int i=4; i<=n; i++)
    {
        a = b + c - (d+1);
        d = c;
        c = b;
        b = a;
    }
    return a;
}
```

7. [6 punti] Scrivere una funzione che riceva in input un array  $v$  di  $lv$  double, un secondo array  $w$  di  $lw$  double, con  $lw \leq lv$ , e un valore di tolleranza  $epsilon$ , anch'esso double, e verifichi se  $w$  è contenuto in  $v$  con un'approssimazione di  $\pm epsilon$  su ogni elemento. Si assuma la correttezza dei dati in input.

**Soluzione:**

```
bool ricercaPattern(const double* const v, const int lv, const double* const w,
                   const int lw, const double epsilon)
{
    int cont;
    for (int i=0; i<=lv-lw; i++)
    {
        cont = 0;
        for (int j=i; j<i+lw; j++)
        {
            if (w[j-i]-epsilon <= v[j] && v[j] <= w[j-i]+epsilon)
                cont++;
            if (cont == lw)
                return true;
        }
    }
    return false;
}
```

**Esame di Informatica**  
(Variante A)

---

8. [8 punti] Scrivere una funzione che riceva in input un intero non negativo  $a$  e un secondo intero positivo  $n$ , allochi un array  $b$ , anch'esso ricevuto in input<sup>1</sup>, di  $n$  elementi interi e converta  $a$  in binario con  $n$  bit memorizzando i bit in  $b$  (avendo cura che la cifra più significativa sia memorizzata nella cella 0 e la meno significativa nella cella  $n - 1$ ). La funzione non effettui alcuna conversione nel caso in cui non fosse possibile convertire  $a$  in binario con  $n$  bit. Si assuma la correttezza dei dati in input.

**Soluzione:**

```
void dec2bin(int a, int*& b, const int n)
{
    if (a > pow(2,n)-1)
        return;

    if (b)
        delete [] b;
    b = new (nothrow) int[n];

    for (int i=n-1; i>=0; i--)
    {
        b[i] = a%2;
        a = a / 2;
    }
}
```

9. [8 punti] Scrivere una funzione che riceva in input una matrice dinamica  $M$  di  $n \times n$  elementi interi e verifichi se esiste una sottomatrice nulla  $m \times n$  (con  $m < n$ ).

**Soluzione:**

```
bool checkNull(double** M, const int n, const int r, const int m)
{
    bool found = true;

    for (int i=r; i<r+m; i++)
        for (int j=0; j<n; j++)
            if (M[i][j] != 0)
                return false;

    return true;
}

bool findNull(double** M, const int n, const int m)
{
    for (int r=0; r<=n-m; r++)
        if (checkNull(M,n,r,m))
            return true;

    return false;
}
```

---

<sup>1</sup>Si noti che la funzione si occupa dell'allocazione dell'array  $b$ . Si faccia attenzione che l'informazione su quale sia l'indirizzo del primo elemento dell'array non vada persa al termine dell'esecuzione della funzione.